**Use Case of User and Manager**
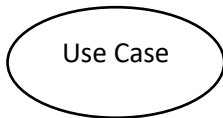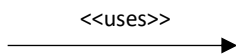
The notation of Use Case

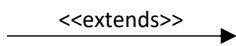Actor

The users of the system

Use Case

The use case of the system
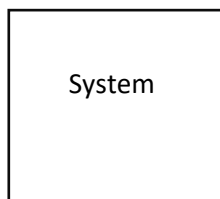
<<uses>>

The include relationship between two use cases

<<extends>>

The extend relationship between two use cases

The extend relationship between two use cases

System

Boundary of System

**Class Diagram**

## Movie

+Movie Name: String

+Movie Show Time: Date/Time

+Venue: Char
___
+Update Details()

## Book Ticket

-No of ticket available: Integer

-Movie Name: Char

-Show No : Integer

-Date : Date Time

-Time : Date Time

-Venue : Char
___
+Update seat: Available()

## Customer

-ID : Char

-Name : Char

-Phone No : Integer

-Address : Char
___
+Sign Up()

+Login()

+Logout()

+Book Ticket()

+Make Payment()

+Cancel Ticket()

+1..*    View    +1..*

+0..*

Updates

+1..*

## Admin

-ID : Char

-Name : Char

-Password: Char
___
+Add Movies()

+Update Movies()

+Delete Movie()

+1..*    Book

+1..*

+1..*

Make Payment

+1..*

## Make Payment

-ID : Char

-Amount : Float

-Customer Id : Char

Transection Id : Integer
___
+Confirm Transection()

**Sequence Diagram**

| Admin | User | Movies | Book Ticket | Payment |
|-------|------|--------|-------------|---------|

Add movie( )

Return status

Update movie( )

Return status

Delete movie( )

Return void

Choose( )

Return status

Login

Booking data( )

Avoiding booking collisions( )

Return status

Cancel Ticket( )

Payment method( )

Confirmation

Log Out